

Philippe Bootz

The Problematic Of Form Transitoire Observable, A Laboratory For Emergent Programmed Art

2005

<https://doi.org/10.25969/mediarep/17665>

Veröffentlichungsversion / published version

Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:

Bootz, Philippe: The Problematic Of Form Transitoire Observable, A Laboratory For Emergent Programmed Art. In: *Dichtung Digital. Journal für Kunst und Kultur digitaler Medien*. Nr. 34, Jg. 7 (2005), Nr. 1, S. 1–16. DOI: <https://doi.org/10.25969/mediarep/17665>.

Nutzungsbedingungen:

Dieser Text wird unter einer Creative Commons - Namensnennung - Weitergabe unter gleichen Bedingungen 4.0/ Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz finden Sie hier:

<https://creativecommons.org/licenses/by-sa/4.0/>

Terms of use:

This document is made available under a creative commons - Attribution - Share Alike 4.0/ License. For more information see:

<https://creativecommons.org/licenses/by-sa/4.0/>

The Problematic Of Form Transitoire Observable, A Laboratory For Emergent Programmed Art

By Philippe Bootz (with the linguistic assistance of Loss Pequeño Glazier)

No. 34 – 2005

Abstract

I will present some conceptions of programmed art focused on the problem of form. I will not explain here the different approaches but only open the question in the perspective of the procedural model. I will start from the basic common point of view of the collective Transitoire Observable and, after an overview of some aspects of the procedural model, I will pose the question of form as a specific management in programming of arbitrary aesthetic constraints that are posed by the author in his management of the situation of communication created by the work whatever the surface aesthetics is on screen. In this sense, we will speak of “programmed forms” as forms in programming and not as forms of the programmed multimedia event.

1. Transitoire Observable And the Question Of A Specificity Of Programmed Forms.

1.1. Programming As A Matter

E-poetry first focused on a programming approach. But since the end of the 90s, many e-poetry works have focused on video art aspects of multimedia events on screen or have explored “traditional” use of the link in hypermedia. Many others use the electronic medium only as a simple medium and as a support for a classical visual poetic approach or as classical visually illustrated texts. These orientations are encouraged by firms that produce software in which the specifics of the numerical medium are masked by video metaphors. I do not deny that these kinds of poetic projects will change traditional understanding of the text. They also produce different and non-classical manners of expression. But it seems to me, and also to several other poets and artists, that this popular approach does not meet the

specific needs of this medium because it does not engage programming. Thus these approaches are unable to propose a situation of communication that is truly new.

This is why, in 2003, on the suggestion of Alexandre Gherban, Alexandre, Tibor Papp and I decided to break with these approaches and to create a new collective that relocates programming at the core of electronic art. We thought that certainly there existed specific forms closed to programming, forms we did not know and had to explore. We wrote a manifesto in which we explained that we consider programming (and not the program) the material of this art, that the multimedia event that appears on screen is only a transient observable state (a “transitoire observable” in French) that occurs while running. We think that forms exist that engage the complexity of the situation of communication created by programmed works. We also think that these forms are close to programming and highly independent of the nature of the transient observable. Programming is a new kind of material that artists can sculpt and model. It is a matter of a formal approach to the algorithms/processes duality. What is pointed out in this approach is the relationship between the algorithmic nature of code and the pragmatics of reading.

So, at this level, e-poetry certainly does not exist but one can gesture towards a general art of programming. This is why several poets and artists quickly joined the collective: Jim Andrews, Wilton Azevedo, Jean-Pierre Balpe, Bluescreen, Michel Brett, Patrick Burgaud, Philippe Castellin, Xavier Leton, Antoine Schmitt, Eric Serandour, and Reiner Strasser.

1.2. Major Directions Of Exploration.

These artists employ numerous different surface aesthetics, generally issued from previous programmed works, often published in *alire*. These works can be interactive or not, but they always put the reader in a particular position where he is responsible for an experiment of reading, and is not the focus of the work: the reader is a partner. He has a special role to play in the work. His reading no longer remains outside of the work. Through its dark moments, its failures, its shortcuts, its questionings, and its great place of non-meeting, the reading activity fully participates in the work, is a process of the work, it is an intrinsic component of the work. Reading is truly a sign, as is the running process that it confronts, even in non-interactive works. The work is not designed to satisfy a reader nor is it to be read in the same manner as a book or a video. It is designed to run because physical running is its *raison d'être*. In *alire12*, interactive works by Burgaud, Strasser, Schmitt, Gherban and Andrews satisfy such an assertion on different levels.

Interactive or non-interactive works are conceived to confront the intentionality of the author and that of the reader. The intentionality of the reader passes through interactivity and expressive non-interactive behaviour. The intentionality of the

author is expressed on screen and in the program, not only in lines of code that generate visible materials or that govern expressive visible behaviors, but in more specific meta-stylistic rules applied as arbitrary constraints on the procedures that manage the observable multimedia transient event.

It appears in *Transitoire Observable* that meta-stylistic rules can be similar in e-poetry and in more general programmed art, it is the ground of the collective. This convergence of points of view asks the question of the existence of specific programmed forms, independent of the surface aesthetical level on screen. It also opens, perhaps, the possibility of a-media multimedia in which meta-rules would be managed in a very high abstract level, without directly taking into account the specificity of media.

2. Main Theoretical Features that Appeared In *Alire*

To explain how these authors arrived at these conceptions, it is useful to know the most important concepts that emerged from works published in *alire*. These conceptions are relative to the problem of meaning and to the relations that exist between signs and author, signs and reader, author and reader. These works have broken new ground on three levels. The second and third level constitute what I will call the “procedural model”.

2.1 The Surface Rhetorical Level

First, *alire* has added some new specific rhetorical possibilities, namely syntactical animation. It has also published other new specific rhetorical possibilities: poetic hypertext, kinetic poetry, and algorithmic poetry (combinatory or automatic generated poetry).

In each case, generally, we only need traditional semiotic theories to explain this surface rhetorical level. For example, all features in syntactical animations can be explained by using a traditional linguistic definition of text and a traditional approach to syntax. We only need to consider the difference between temporal and spatial construction of signs inside the same set of words and letters (spatial reading and temporal reading).

Unfortunately, this level fails to explain most of significant features in interactive works. It also fails to explain the changing of meaning with time or the difference between combinatory and automatic generation. So, we need to consider the totality of the situation of communication between the author and the reader as constituting a system. This communication passes through the work. Writing and

reading are internal functions of this system, which constitutes the second level of signs.

2.2 the Communicative Level

a) Cognitive and technical understanding.

At this level, we cannot neglect the technical behavior of the system in the construction of meaning. Communication features, ergodic activity of reading, semiotic features, cognitive features and technical features cannot be disjunctive. Notably, *the reader cannot construct a meaning of the work without constructing at the same time a meaning of the totality of the situation of communication.* From a systemic point of view, the work has no reality. It can only be understood as a subsystem of the situation of communication that cannot be extracted from this situation. Two important and not classical rhetorical features (namely the double reading and the aesthetic of frustration) use this relation between construction of the meaning of the work and the construction of the meaning of the situation of communication.

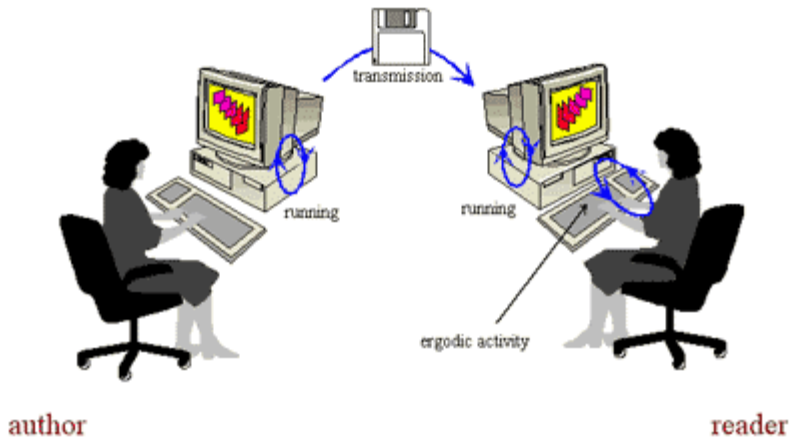


Figure 1: system's structural schema

It is useful to consider the cognitive level because the archetypal mind representations that are used by a reader are not stable, they are based on experiments of reading or writing in his particular medium and do not come from a deep cultural tradition. Thus they can greatly differ from one person to another. This mind representation has been called the "profondeur de dispositif" (system depth) in the procedural model. A system depth is applied as a filter by the cognitive

strategy of reading or writing in order to decide what is the significant of the signs in the transient observable event that is produced while running. We can find many relations between different works in *alire* that show that a number of these pieces, independently of the nature of the work, have constructed the actual mind representation (which is described by the procedural model) by using in fact exactly the same strategy to manage reading.

At a technical level, author and reader are only users of the computer. Notably, the author does not manage in his engagement the totality of the rules that are used by the computer while running. Hence the computer no longer is a Turing machine. Another model must be used. I have constructed the model of the “machines de monstration” (machines of showing). The program written by the author does not totally manage the physical process of running. That is, the algorithmic level of the program is not completely responsible for the functioning of the transient observable. We can say that the author is the author of the program and data, but only the co-author of the physical process that appears to the reader while the machine is running. Using more traditional literary language, we can say that the author's program contains a large level of the “non-said”. But this non-said does not play the same role as the non-said of the classical printed text: this non-said will be interpreted by the machine and not by the reader to produce the observable sign. Thus it is necessary to consider that program (algorithmic level) and physical process while running are two different and complementary parts of the work. Adaptive generation uses this unsaid as a constraint for programming.

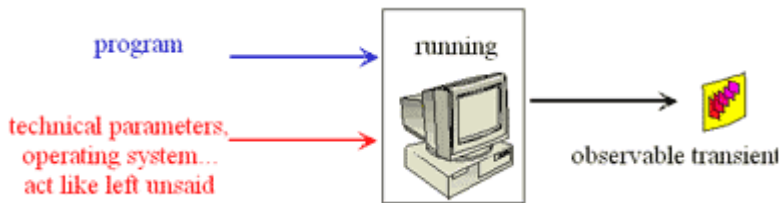
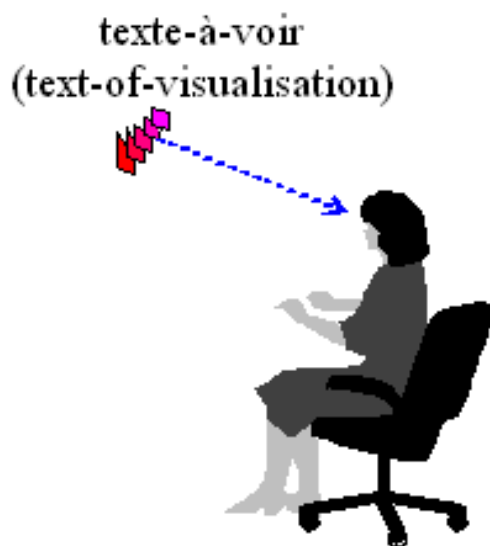
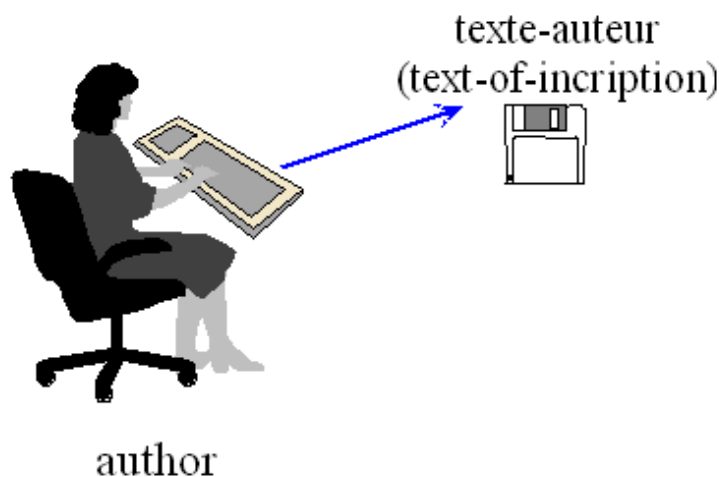


Figure 2: Technical left unsaid

b) Semiotic and communicative understanding

From a semiotic point of view, we can separate the classical and general semiotic notion of text (the text is the object of the interpretation) into three different parts that do not act in the same space. Program and data constitute the “texte-auteur”. This is a sign that is only accessible by the author. It is in the domain of the author. The second sign is constituted by what will be considered as “the text” by the reader. It is the “texte-à-voir” (text-to-be-seen). It is a part of the observable transient event that can differ from a reader to another because readers will not apply the same system depth on the transient observable. The physical process itself is a function.

From a semiotic point of view, it transforms the "texte-auteur" into the "texte-à-voir". Because it generates the transient observable, it is called "generation" in the model.



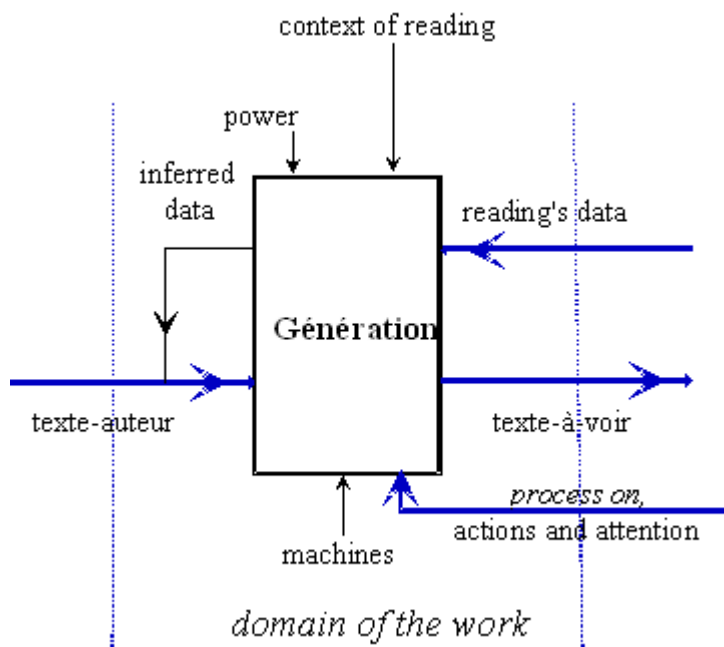


Figure 3: The different structural and functional components of the work

The totality of this subsystem constitutes the “domain of the work”. It plays the role of an interface between the domain of the author and the domain of the reader. Thus it is the part of the model that is most similar to the classical notion of work. A work can no longer be regarded as an object, but as a complex set of components of different natures (object for the *texte-auteur*, state for the *texte-à-voir* and process for the generation) that can only exist while the program is running.

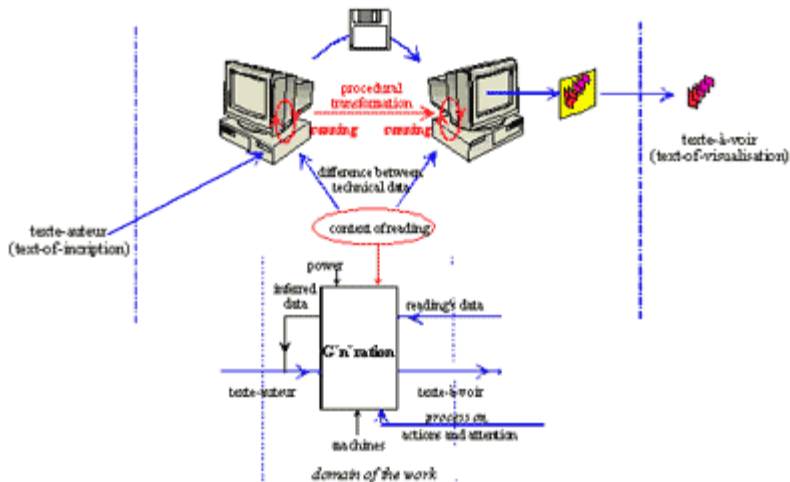


Figure 4: Structural and functional schemas of the domain of the work

This set of signs must be completed by two mind representations that are the classical meanings of the work and the totality of the situation of communication. These two mind representations are the "texte-écrit" and the "texte-lu". The "texte-écrit" is used by the author to construct the *texte-auteur* and, notably, to elaborate strategies of temporal, space, and media behaviors on screen, to delimitate what he will consider as interface and what he will consider as *texte-à-voir* in the transient event, to manage interactivity.

As he is inside the system of communication, the reader can give meaning to his own ergodic activity while he is interacting. This special meaning is called "double reading". Generally, it arrives during reading to reorient in another way the construction of the work's meaning. Double reading uses the activity of reading as a sign intended for the reader. In double reading, interactivity acts as a reflexive sign: the reader interacts with himself. Double reading has been used since the beginning of the 90s.

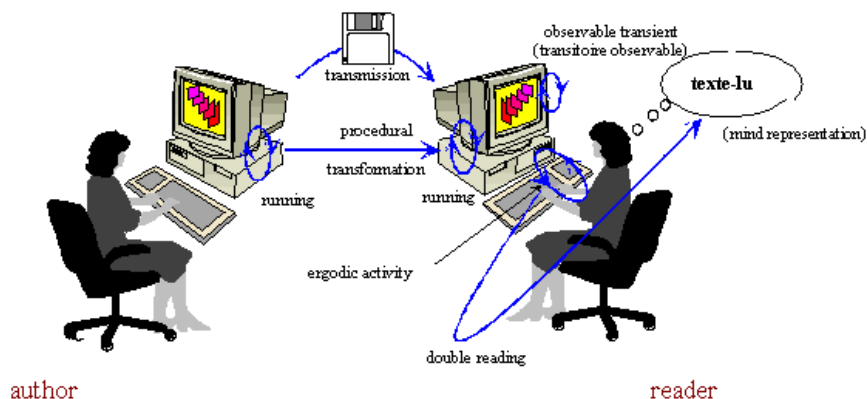


Figure 5: The double reading.

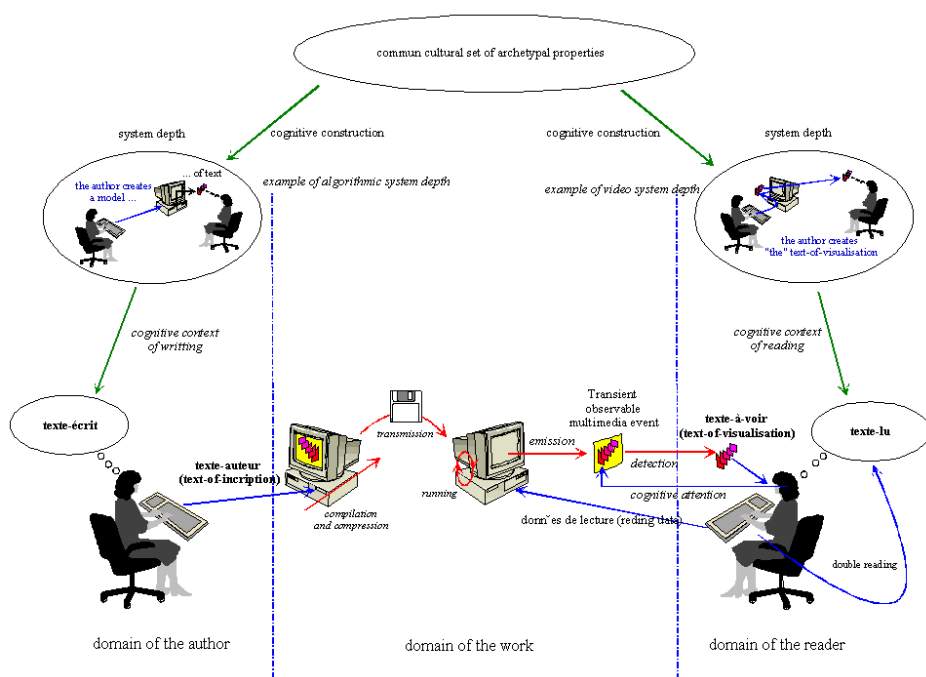


Figure 6: Complete description of the situation of communication in an example where the author has an algorithmic system depth and the reader a video system depth.

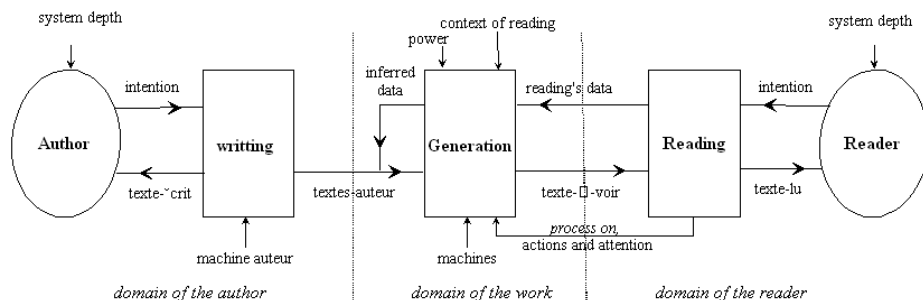


Figure 7: Functional scheme of the situation of communication.

2.3. The meta-level

The theory of double reading has shown that reading can act as a sign. When it does, for example in the aesthetic of frustration, this sign is a part of the situation of communication and not of the domain of the work. Thus the totality of the situation can become a sign, a meta-work. It is an iconic sign of a conceptual reality of the work that does not match to the surface rhetorical level. The referent of this sign is usually a situation or a process in life. The meta-work is iconic in the situation it creates or in its functioning. This meta-work is intended for a person watching the reader while he is reading. This person is not reading himself but watching somebody else reading. He is a meta-reader. Reading and meta-reading do not apply to the work by the same canal. They are two different functions applied to the work, even if they can be successively applied by the same person. The meta-reader can understand the reactions and actions of the reader in a certain manner that differs from the reader's double reading. But he can only do this if he knows the meaning the author gave to the reading activity. This meaning is expressed in papers or other paratexts¹. It is not expressed at all in the *texte-auteur*. Some features in the program or other data of the *texte-auteur* can only play the role of an index for this meta-level of author's intentionality. Meta-reading is not an interpretation of the situation but a recognition of it. The meta-reader must also perceive the specific influence of program non-said while running. Thus meta-reading can be seen as "a most intellectual understanding" of what happens in the situation of communication whereas reading can be regarded as "a most affective understanding" of what happens in the work. For double reading, the activity of reading is a significant level of the communication between the author and the reader. For meta-reading, the activity of reading is an expressive level of the communication between the author and the meta-reader².

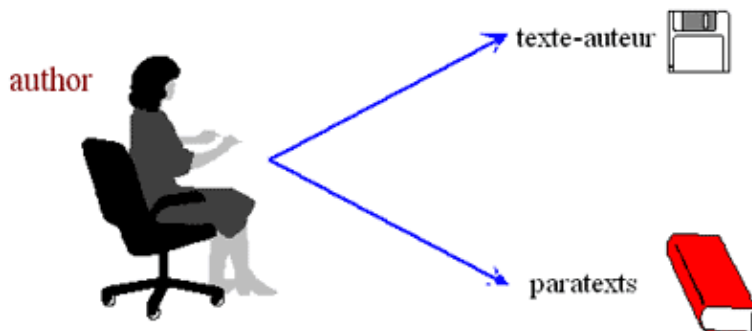


Figure 8: Total activity of the author.

Figure 8: Total activity of the author.

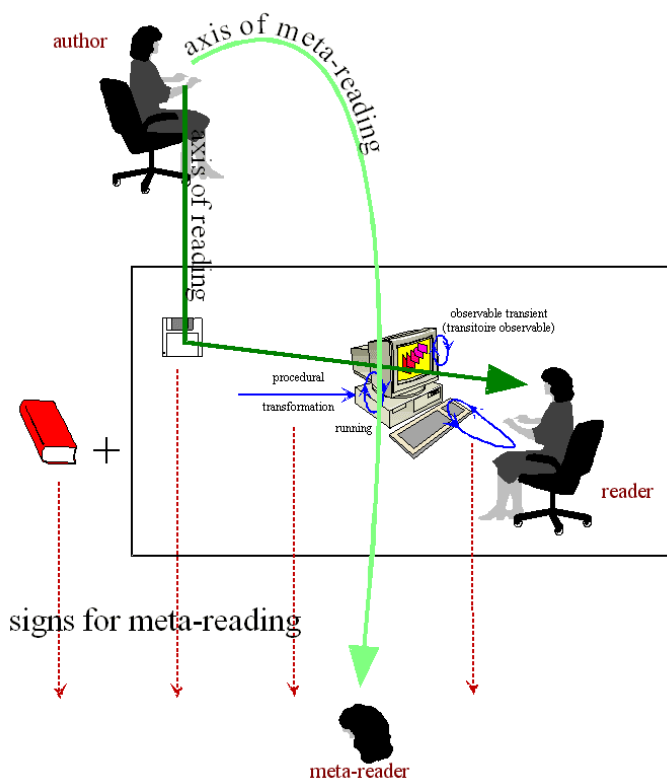


Figure 9: Difference between reading and meta-reading.

The aesthetic of frustration gives a specific role to reading and considers that the reader is not the only person addressed by the work. The reader plays a specific, unexplained role through reading because reading is the scene of this role. Expressive reactions, actions, are signs. They are in an iconic relation to real life activities. This aesthetic considers that traditional failure of reading or rereading is a specific modality of reading, it has a specific meaning at the scene of activity. Thus the work can be very frustrating for the reader but, from the point of view of meta-reading, a failure of reading cannot exist. Even a failure produces meaning but this meaning doesn't pass through a reader's understanding of the work when reading fails.

When using meta-reading, e-poetry is fundamentally concerned with communication by natural language and by the real nature of reading and writing in a society of communication. Many works argue that the cultural basis of the society of information and communication is a myth. These works are based on miscommunication. They are using the most mythic device of communication, the computer, because it is fundamentally a machine of language, a machine of programmed language, whatever the nature of signs on the surface of the screen is.

3. The Question of form in programmed e-poetry.

3.1 Different levels of constraint.

The question of form in this system is relative to the role that the author gives to each part of the domain of the work. It is fundamentally relative to the system depth of the author. It depends only on the author, not on the reader. I define a form as a creation and a management of an arbitrary set of constraints. It is an efficient point of view because we can find different levels of constraints by taking into account the relationship between the different components of the domain of the work.

3.2 Simulation and intersemiotic.

A first set of constraints is only relative to the surface rhetorical level. There are two possibilities nowadays: the observable transient simulates a traditional form (from traditional or visual poetry) or it mixes different semiotic behaviors in an interpoetic (or intersemiotic) physical behavior. Intersemiotic works can confront different kinds of observable signs (for example in video poetry) or can be a transformation with time of a sign into another from the same semiotic (a manner explored by Jacopo Baboni Schilingi in music).

When the author only uses these kinds of constraints, the computer is used as a tool, an amplification of complexity and the only question of form that remains to be solved by the author, is the transformation of the observable transient to a *texte-à-voir*. The other components of the domain of the work do not play any role at all. These kinds of system enter in the class of “computer assisted literature” and not in the class of *Transitoire Observable*’s “programmed literature”. In programmed literature, the aesthetic problem of this pure observable level is the same as in computer assisted literature, it concerns the rhetorical level, but the general problem of form does not restrict to this level.

For instance, temporal programming uses the transformation of signs of the same semiotic system. In this approach, the beginning and the end of the transformation can be recognized as a certain sign (or set of signs) by the reader, but the transient part is only interpreted by a cognitive interpolation with the beginning or the end. It seems that this part uses a specific temporal semiotic that preserves the global coherence of reading. This temporal semiotics is not interpreted in the same semiotic. It is why this transformation has special properties for the reader: it occurs as a continuous physical process but a semiotic discontinuity, a step between two non-simultaneous interpretations, and, while the physical process is linear and limited, the cognitive interpretation is looping between the two interpretations: they finally coexist in mind. This constraint can be described in two rules: beginning and end of the sequence must be recognizable signs in a certain semiotic and the transformation time must be a physically continuous temporal event. This event, probably, is a unique UST³.

3.3 Using meta-stylistic rules.

The communicative and meta level give types of constraints that are more specific to programmed art because they use real time running while reading. The aesthetic of frustration or the strategies of managing of the reading I have presented above concern one type of constraint: how to manage the interactivity? In contrary to the hypertext philosophy, we see that the reader’s intentionality is not the focus of the work. In *Transitoire Observable*, the old adage introduced in *alire* in the eighties always applies: reading forbids to read.

Another set of constraints is based on the relationship between the *texte-auteur* and the *texte-à-voir* and concerns the generative level of the work. In this relation, the *texte-auteur* is a model of the *texte-à-voir*. This point was discussed by Jean-Pierre Balpe in his theory of the meta-author⁴. Historically, the first kind of programmed texts using this conception was automatic generation. But adaptive animation made by L.A.I.R.E. also obeys to this conception, even if the observable behavior is very different. The difference is due to the nature of the algorithms that are used. Automatic generation uses algorithms of creation of linguistic material although

programmed animation uses algorithms to manage the expressive aesthetic behavior of the observable material. But this difference only concerns the transformation of the *texte-auteur* into the *texte-à-voir*, it does neither concern programming itself nor the conception of model. In each case, the author has to write two different kinds of rules: rules of the semiotic level of the *texte-à-voir*⁵ and rules of management of these rules. This second set of lines of code cannot be seen by the reader. Adaptive⁶ generation shows that it becomes today a model of models. Thus these works manifest two aesthetic levels. First, is classically showing the reader the transient observable through observable media, processes, and behaviours. The second is only indexed in the observable transient. To perceive it, it is necessary to re-run the work, sometimes on another computer⁷. This second level is the manifestation of meta-stylistic rules. Because they do not directly appear to the reader, we can be sure that the reader is no more, through the operation of reading, the addressed person of the work. In fact, the real addressed person of the work is a social and cultural meta-reader. A given person, as he is an instance of this cultural meta-reader, can play the two roles but they cannot be assumed at the same time: meta-reading does not occur through the reading of the work. There exists a real rhetorical unity between interactive works that use the aesthetics of frustration and those that are not interactive but use meta-stylistic rules.

Generally, the meta-stylistic rules take the form of parameters, or logical set of rules, for dialogue between parts of the program, or special non-visible behaviours like measures. These meta-stylistic rules are pure arbitrary rules of creation. They are not justified by technical features. Thus, they are really artistic constraints. When meta-stylistic rules have been choice, the problem to manage them consists to construct a impressive tool to describe (conception level) and to program them. For complex projects the conception level of the constraint is the most difficult. Authors in *Transitoire Observable* use different models of conception: Jean-Pierre Balpe uses the system of generation he has created, Tibor Papp uses a grammar in which paradigms are individual behaviours and syntax a set of rules of assembling, Alexandre Gherban uses also a set of behaviours (temporal behaviours), but these behaviours are choosen and parametrized by using cellular automates, Antoine Schmitt uses physical mechanical laws and a predictive calculus on temporal specific objectives (aesthetic of causality), Xavier Leton and I use measures and rules of adaptation, I use a generative editing (the horizontal editing) to construct observable behaviours with elementary components, Bluescreen uses arbitrary laws of interaction between programmed objects to construct a collective behaviour...

To cite examples from *alire*¹², the visible processes can be managed by algorithms of artificial life (Gherban, Bluescreen), by physical models (Schmitt), or by temporal programming (Bootz). But, in each case, the transient observable is not understood as a simulation. If life exists, it does not occur in the result but in the principle of the

program's functioning. It is not biological life that would obey the laws of duplication and competition, but an aesthetical life in which these algorithmic rules are constrained by meta-aesthetical rules that do not directly appear in the transient observable event. Aesthetical forms at the surface of the screen are instances of forms produced by these meta-aesthetical rules. To do this, the program can measure while running (Bootz). The program uses the results of these measures in order to modify its one strategy. In this way it is an adaptive generator. For example, the result of a calculation by an algorithm can be changed by the result of a measure to obey a certain aesthetical idea, or the measure can forbid certain parts of the program to run. More generally, the program is constructed as a set of independent rules that have a relationship while running in order to construct an observable transient state constrained by aesthetical meta-rules. It is the case, even if the program does not use a specific algorithmic model (Papp). The author does not exactly know what will happen, but the author knows the classes of aesthetical main features that will be made manifest in the observable multimedia transient event when technical unsaid does not trouble programmed process.

Conclusion

Thus, even if each author explores different modalities of meta-rules, with sometimes very different objectives, I think that the existence of these meta-rules is sufficient to speak of a common aesthetic. The common direction for the author is to search for the best form of conception and realisation of the class of meta-rules he explores. This unity is not manifested at screen, but in the similitude in the management of the role of the reader and in the role of the author: it is not an aesthetic of text, it is an aesthetic of creating.

Notes

1. The role of paratext in the interpretation of an electronic artistic work was studied on several examples in the paper *Approche du rôle de la mémoire dans la conception et la réception de quelques œuvres en littérature numérique* for a seminar in Montreal in 2003, to be published in the *revue discours social/social discourse*.
2. See the paper *Approche sémiotique d'un certain art programmé: œuvre signe et méta-lecture*, conference of Cerisy, July 2004, to be published.

3. A Temporal Semiotic Unit. The theory of UST was created in the laboratory of computer music of Marseilles (MIM) in order to analyze music (<http://www.labo-mim.org>). Its extrapolation to visual and multimedia art is today explored. See the paper by Hautbois, Xavier: *Les Unités Sémiotiques Temporelles: de la sémiotique musicale vers une sémiotique générale du temps dans les arts*. For the conference ICMS8, September 2004, to be published.
4. Balpe, Jean-Pierre: *Un roman inachevé – dispositifs*. Littérature n° 96, 1994, pp. 37-53 and Balpe, Jean-Pierre: *Méta-auteur*, alire10, 1997, pp. 95-99.
5. In this case, it is the texte-à-voir for the author, not for the reader. The author manages what happens on the screen of his computer, the semiotic behavior of what he considers on this screen as the texte-à-voir.
6. An adaptive generator tries to manage the technical left unsaid. It changes the logical set of rules used to construct the observable transient by interpreting physical measures made by the program during running.
7. Adaptive generation is totally transparent on a computer because the measures it makes give always the same results. Thus, the changing in the logic of running is the same from a re-reading to another.